

CLEAN VERSION OF REPLACEMENT PARAGRAPHS
TO THE SPECIFICATION

Replacement paragraphs to the specification are shown in this section for purposes of clarity.
The marked up version of the specification is shown at pages 7-10 of this Reply.

Please replace the 10th paragraph (lines 28-29) on page 5 with the following:

A1 Fig. 9 is a flowchart illustrating one example method employed in producing a list in accordance with an aspect of the present invention.

Please replace the 6th paragraph (lines 15-16) on page 6 with the following:

A2 Fig. 16 is a flowchart illustrating a method associated with generating a final list, in accordance with an aspect of the present invention.

Please replace the 2nd paragraph (lines 4-10) on page 20 with the following:

A3 Fig. 9 is a flow chart that illustrates one method employed with the present invention. At 910, processing associated with the artist occurs, while at 920, processing associated with a track occurs.

The following pseudocode implements the example method for identifying an item that a user desires to add to a media library. Identifying the user item to be added to the media library involves matching identifying data associated with the user item to identifying data associated with items already identified by an identifying system. Since the identifying data associated with the user item may not be entirely consistent with identifying data associated with items already processed into the media library, inexact matching of identifying data may be required.

Please insert the following after the 3rd paragraph (ending at line 15) on page 20:

For each reference track

If artist id not same as previous reference track then

Clear user song artist votes

Split artist name into words, canonicalize words

For each artist word

Increment user song artist vote for user song artist names
that contain artist word

End for

For each user that received at least one artist vote

 Compute artist matching score

 If artist matching score is above threshold and artist matching

 score is better than previous artist matching score, then

 artist id of user song = artist id of reference track

 end if

End for

End if

Select user songs whose track name matches first N characters of reference track name

If no user songs match, then return, endif

Split reference track name into words

canonicalize words

clear user song track votes

for each track word

 increment user song track vote for user song track names that contain
 track word

end for

for each user song that received at least one artist vote

 add user song artist vote to user song track vote

end for

for each user song that has at least one track vote

 compute track matching score

 if track matching score is above threshold and track matching score is
 better than previous track

 matching score user song track id = track id of reference track

 end if

end for

End for

A4
cancel.

Please replace the 1st paragraph (lines 1-23) on page 27 with the following:

Fig. 16 is a flowchart illustrating a method 1600 for generating a final list. The following segment of pseudocode can be employed to implement the method illustrated in the flowchart associated with generating a final list *via* generating similarity values. At 1610, the similarity between the Cartesian product of a seed list and a user list is computed. At 1620, the mean similarity over that Cartesian product is computed. At 1630, the similarity list is generated and sorted.

If seed song has track id then

Seed list = seed song

Else if seed song has artist id then

Seed list = all user songs with same artist id

Else

Final list = all user songs with same artist name as seed song

Return final list

Endif

For each user song with track or artist id

If user song has track id then

User list = user song

Else if user song has artist id then

Check cache for similarity of seed to artist id

If similarity value is in cache

Add user song and similarity to final list

Go to next user song

End if

User list = all user songs with same artist id

End if

Compute limits to similarity computation

Compute similarity over all possible pairs of songs on seed list and

user list, subject to limits

similarity value = mean value over pairs

A5
omit

```
        if use song has no track id
            store similarity in cache
        end if
        add user song and similarity to final list
    End for

    Sort final list by decreasing similarity
    Return final list
```

A5
Cmld.

In the case where the seed song has a known track identifier, then the seed list consists of the seed song. Otherwise, if the seed song has a known artist identifier, then the seed list consists of user items whose artist identifier matches the seed song artist identifier. Otherwise, if the seed song is unknown, the final similarity list will include those user items whose artist name matches the seed song artist name. If there is more than one seed song, the seed list can be the concatenation of the seed list for all of the individual songs that have known artist or track identifiers. A user list is computed analogously for user songs that contain a track identifier or artist identifier. If the user song has a track identifier, then the user list is the user song. Otherwise, the user list is user songs whose artist identifier matches the artist identifier of the user song under consideration. For efficiency concerns, in one example of the present invention, limits may be placed on the Cartesian product between the seed list and the user list. If one of the two lists contains only a single song, iteration over the other list is limited to a pre-determined, configurable threshold (*e.g.*, to 200 items). If both lists contain more than one song, then iteration over both lists is similarly limited to a pre-determined, configurable threshold (*e.g.*, to 30 items). For further efficiency concerns, in one example of the present invention, when a user song does not have a track identifier, the similarity between the seed song(s) and the user song is cached, so that if another user song has the same artist identifier but a missing track identifier, then the cached value may be retrieved and employed.
